UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/629,160	07/29/2003	Jerome Huck	10006029-2	5990

7590 06/22/2007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
VICARY, KEITH E	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
06/22/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/629,160	HUCK ET AL.	
	Examiner	Art Unit	
	Keith Vicary	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 February 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-6,9,12,14,15,17,18 and 20-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-6,9,12,14,15,17,18 and 20-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 July 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *7/29/2003*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-2, 4-6, 9, 12, 14-15, 17-18, and 20-24 are pending in this office action and presented for examination. Claims 23-24 have been newly added and claim 14 has been amended by amendment filed 2/17/2004.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 1-2, 4-6, 9, 12, 15, and 17 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-4 and 11-12 of U.S. Patent No. 6643769. Although the conflicting claims are not identical, they are not patentably distinct from each other because each of these claims in the instant

application has a corresponding claim in the '949 application which contains all elements of the instant claim. (In re Goodman 29 USPQ2d 2010 (Fed. Cir. 1993).

4. Claim 1 of the '769 patent contains every element of claim 1 of the instant application.

5. Claim 2 of the '769 patent contains every element of claim 2 of the instant application.

6. Claim 3 of the '769 patent contains every element of claim 4 of the instant application.

7. Claim 4 of the '769 patent contains every element of claim 5 of the instant application.

8. Claim 1 of the '769 patent contains every element of claim 6 of the instant application.

9. Claim 1 of the '769 patent contains every element of claim 9 of the instant application.

10. Claim 1 of the '769 patent contains every element of claim 12 of the instant application.

11. Claim 11 of the '769 patent contains every element of claim 15 of the instant application.

12. Claim 12 of the '769 patent contains every element of claim 17 of the instant application.

13. Claims 14, 18, and 20-24 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 2 and 10 of U.S. Patent No. 6643769. Although the conflicting claims are not identical, they are not patentably distinct from each other because claims 14, 18, and 20-24 of the instant application are obvious variants of claims 2 and 10 of the '769 patent.

14. Claim 14 and 23-24 of the instant application are obvious variants of claim 10 of the '769 patent, as the system embodiment of claims 14 and 23-24 are obvious given the method limitations of claim 10.

15. Claims 18 and 20-22 of the instant application are obvious variants of claim 2 of the '769 patent, as the method steps of claims 18 and 20-22 are obvious given the system limitations of claim 2.

Claim Objections

16. Claims 1 and 14-15 are objected to because of the following informalities. Appropriate correction is required.

a. Claims 1 and 14-15 recite the limitation "run time data." When used as an adjective, "run-time" should be hyphenated. See "http://foldoc.org/?query=runtime" for details.

Claim Rejections - 35 USC § 112

17. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

18. Claims 2 and 14-22 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

19. Claims 2 and 14-15 recite the limitation "said program" in lines 2, 6, and 6 respectively. There is insufficient antecedent basis for this limitation in the claim.

b. Claims 17-18 and 20-24 are rejected for failing to alleviate the rejection of claims 14 and 15 above.

20. Claim 14 recites the limitation "said memory addresses" in line 10. There is insufficient antecedent basis for this limitation in the claim.

c. Claims 23 and 24 are rejected for failing to alleviate the rejection of claim 14 above.

21. Claims 20 and 21 recite the limitation "said set of instructions" in lines 3-4 and 3 respectively. There is insufficient antecedent basis for this limitation in the claim.

d. Claims 21 and 22 are rejected for failing to alleviate the rejections of claims 20 and 21 above.

22. Claim 22 recites the limitation "the step of testing" in line 1. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 102

23. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

24. Claims 1-2, 4, 6, 9, 12, 14-15, 17-18, and 20-24 are rejected under 35 U.S.C. 102(e) as being anticipated by Kasper (US 6691308 B1).

25. Consider claim 1, Kasper discloses memory for storing a computer program (Figure 1, ROM 14, described in col. 1, line 57, and cache 20, described in col. 2, lines 58-60, together comprise memory), said computer program having a set of code and a branch instruction (col. 1, lines 56-58, plurality of instructions; note that nothing in the disclosure prohibits any instruction including a branch instruction from having code inserted before it); and processing circuitry configured to receive run time data indicating whether said set of code is enabled and to set a value of a mode indicator based on said run time data (Figure 1, cache control register 23; col. 4, lines 8-10, the CCR 23 allows one to control the cache unit and to control insertion of an instruction into the pipeline 24; col. 4, lines 22-23, the mask register provides flexibility to the circuit; col. 4, lines 11-15, by setting various bits in the cache control register 23, one would be able to enable/disable the cache unit...and control the general operation of the cache unit 20. Note that col. 3, lines 58-65 disclose that contents of the cache can be downloaded by the host system, but does not disclose anything about the mask register or cache control register being downloaded by the host system; thus the mask register and cache control register are user accessible by a typical write instruction as in col. 4, lines 32-34), said processing circuitry configured to receive and execute said branch

instruction during a run of said computer program (see, for example, Table 1, code stream; although a branch instruction is not explicitly disclosed in the code stream; a branch instruction is nevertheless a type of instruction, and Kasper's invention works regardless of the instruction type), said processing circuitry configured to branch to a first address of said memory (if a branch instruction is encountered and the cache unit is enabled as disclosed above regarding the bits of the cache control register, and its opcode or address at which it resides matches a value in the cache, and the control flag code as described in col. 3, line 15-16 has the "I" or "M" flag enabled as in col. 3, lines 30-35, the processor will in effect branch to that address in the cache memory, which contains the op-code to be inserted into the code stream), in response to said branch instruction and based on said value of said mode indicator when said set of code is enabled (col. 4, line 15 as disclosed above, by setting various bits in the cache control register 23, one would be able to enable/disable the cache unit), said processing circuitry further configured to branch to a second address of said memory (when the cache unit is disabled, no additional code is inserted and the instruction is completed as normal, such as in col. 5, lines 66-67 and col. 6, lines 1-4), in response to said branch instruction and based on said value of said mode indicator, when said set of code is disabled (col. 4, line 15 as disclosed above, by setting various bits in the cache control register 23, one would be able to enable/disable the cache unit).

26. Consider claim 14. Kasper discloses means for storing a computer program (Figure 1, ROM 14, described in col. 1, line 57, and cache 20, described in col. 2, lines

58-60, together comprise memory), said computer program having a set of code and a branch instruction (col. 1, lines 56-58, plurality of instructions; note that nothing in the disclosure prohibits any instruction including a branch instruction from having code inserted before it), said branch instruction including an address identifier identifying a first memory address (col. 4, lines 5-8, when there is a comparison match, the instruction from the memory unit 14 is replaced by a new instruction from the cache unit 20 in the pipeline 24; the first address is the address in the cache that is inserted into the code stream; the address identifier of the branch instruction is either the opcode of the branch instruction or the address at which it resides, as either of these values is matched to the cache and thus identifies the line of the cache at which the replacement/inserted instruction lies); means for receiving, during a run of said program, run time data indicating whether said set of code is enabled (Figure 1, cache control register 23; col. 4, lines 8-10, the CCR 23 allows one to control the cache unit and to control insertion of an instruction into the pipeline 24; col. 4, lines 22-23, the mask register provides flexibility to the circuit; col. 4, lines 11-15, by setting various bits in the cache control register 23, one would be able to enable/disable the cache unit...and control the general operation of the cache unit 20; Note that col. 3, lines 58-65 disclose that contents of the cache can be downloaded by the host system, but does not disclose anything about the mask register or cache control register being downloaded by the host system; thus the mask register and cache control register are user accessible by a typical write instruction as in col. 4, lines 32-34); means for setting a value of a mode indicator based on said run time data (Figure 1, cache control register 23; col. 4, lines 8-

10, the CCR 23 allows one to control the cache unit and to control insertion of an instruction into the pipeline 24; col. 4, lines 22-23, the mask register provides flexibility to the circuit; col. 4, lines 11-15, by setting various bits in the cache control register 23, one would be able to enable/disable the cache unit...and control the general operation of the cache unit 20); means for identifying a second memory address in response to said branch instruction (it is inherent that a branch instruction includes some sort of address identifier to which the program flow will branch to); means for selecting one of said memory addresses in response to said branch instruction and based on said value of said mode indicator (if a branch instruction is encountered and the cache unit is enabled as disclosed above regarding the bits of the cache control register, and its opcode or address at which it resides matches a value in the cache, and the control flag code as described in col. 3, line 15-16 has the "I" or "M" flag enabled as in col. 3, lines 30-35, the processor will in effect branch to that address in the cache memory, which contains the op-code to be inserted into the code stream); and means for branching to said selected address in response to said branch instruction (Figure 1, for example, shows the cache 20 connected to a mux 16 from which an instruction from the cache enters the pipeline), said branching means including a means for executing an instruction at said selected address (Figure 1, execute block in pipeline).

27. Consider claim 15, Kasper discloses storing a computer program in memory (Figure 1, ROM 14, described in col. 1, line 57, and cache 20, described in col. 2, lines 58-60, together comprise memory), said computer program having a set of code and a

branch instruction (col. 1, lines 56-58, plurality of instructions; note that nothing in the disclosure prohibits any instruction including a branch instruction from having code inserted before it), said branch instruction including an address identifier identifying a first address in said memory (col. 4, lines 5-8, when there is a comparison match, the instruction from the memory unit 14 is replaced by a new instruction from the cache unit 20 in the pipeline 24; the first address is the address in the cache that is inserted into the code stream; the address identifier of the branch instruction is either the opcode of the branch instruction or the address at which it resides, as either of these values is matched to the cache and thus identifies the line of the cache at which the replacement/inserted instruction lies); receiving, during a run of said program, run time data indicating whether said set of code is enabled (Figure 1, cache control register 23; col. 4, lines 8-10, the CCR 23 allows one to control the cache unit and to control insertion of an instruction into the pipeline 24; col. 4, lines 22-23, the mask register provides flexibility to the circuit; col. 4, lines 11-15, by setting various bits in the cache control register 23, one would be able to enable/disable the cache unit...and control the general operation of the cache unit 20; Note that col. 3, lines 58-65 disclose that contents of the cache can be downloaded by the host system, but does not disclose anything about the mask register or cache control register being downloaded by the host system; thus the mask register and cache control register are user accessible by a typical write instruction as in col. 4, lines 32-34); setting a value of a mode indicator based on said run time data (Figure 1, cache control register 23; col. 4, lines 8-10, the CCR 23 allows one to control the cache unit and to control insertion of an instruction

into the pipeline 24; col. 4, lines 22-23, the mask register provides flexibility to the circuit; col. 4, lines 11-15, by setting various bits in the cache control register 23, one would be able to enable/disable the cache unit...and control the general operation of the cache unit 20); identifying a second address in said memory and in response to said branch instruction (it is inherent that a branch instruction includes some sort of address identifier to which the program flow will branch to); branching to said second address based on said value of said identifying step and said value of said mode indicator (when the cache unit is disabled, no additional code is inserted and the instruction is completed as normal, such as in col. 5, lines 66-67 and col. 6, lines 1-4); and executing an instruction at said second address in response to said branching step (Figure 1, execute block in pipeline).

28. Consider claims 2 and 22, Kasper discloses said set of code, when executed by said processing circuitry, tests for errors that occur during said run of said program (col. 1, lines 26-32, debug circuit; col. 2, lines 34-37, instruction content...may be corrected, modified and/or debugged; col. 5, lines 13-20, status buffer 34 of Figure 2 stores debug information).

29. Consider claims 4 and 17, Kasper discloses said processing circuitry is further configured to maintain said value of said mode indicator during said run of said computer program and until termination of said run (col. 4, lines 8-10, the cache control register allows one to control the cache unit and enable/disable the cache unit; given

that it allows one to control the cache unit, it is inherent that if one does not choose to control the cache unit, the value will be maintained).

30. Consider claim 6, Kasper discloses said branch instruction includes an address identifier that identifies said first address, and wherein an instruction of said set of code is stored in said memory at said first address (col. 4, lines 5-8, when there is a comparison match, the instruction from the memory unit 14 is replaced by a new instruction from the cache unit 20 in the pipeline 24; the first address is the address in the cache that is inserted into the code stream; the address identifier of the branch instruction is either the opcode of the branch instruction or the address at which it resides, as either of these values is matched to the cache and thus identifies the line of the cache at which the replacement/inserted instruction lies).

31. Consider claim 9, Kasper discloses said branch instruction includes an address identifier that identifies said second address, and wherein an instruction of said set of code is stored in said memory at said first address (it is inherent that a branch instruction includes some sort of address identifier to which the program flow will branch to; if a branch instruction is encountered and the cache unit is enabled as disclosed above regarding the bits of the cache control register, and its opcode or address at which it resides matches a value in the cache, and the control flag code as described in col. 3, line 15-16 has the "I" or "M" flag enabled as in col. 3, lines 30-35, the processor

will in effect branch to that address in the cache memory, which contains the op-code to be inserted into the code stream).

32. Consider claim 12, Kasper discloses said branch instruction includes an address identifier identifying one of said addresses (it is inherent that a branch instruction includes some sort of address identifier to which the program flow will normally branch to).

33. Consider claim 18, Kasper discloses said branch instruction includes an address identifier identifying said second address (it is inherent that a branch instruction includes some sort of address identifier to which the program flow will normally branch to), said identifying step including the step of selecting said second address identifier based on said value of said mode indicator; when the cache unit is disabled, no additional code is inserted and the instruction is completed as normal, such as in col. 5, lines 66-67 and col. 6, lines 1-4).

34. Consider claim 20, Kasper discloses said instruction executed in said executing step is outside of said set of code, and wherein said branching step prevents execution of said set of code, said value of said mode indicator indicating that said set of instructions is disabled (when the cache unit is disabled, no additional code is inserted and the instruction is completed as normal, such as in col. 5, lines 66-67 and col. 6, lines 1-4).

35. Consider claim 21, Kasper discloses said instruction executed in said executing step is included in said set of code, said value of said mode indicator indicating that said set of instructions is enabled (if a branch instruction is encountered and the cache unit is enabled as disclosed above regarding the bits of the cache control register, and its opcode or address at which it resides matches a value in the cache, and the control flag code as described in col. 3, line 15-16 has the "I" or "M" flag enabled as in col. 3, lines 30-35, the processor will in effect branch to that address in the cache memory, which contains the op-code to be inserted into the code stream).

36. Consider claim 23, Kasper discloses said selecting means is configured to select said first memory address in response to said branch instruction if said mode indicator has a first value (if a branch instruction is encountered and the cache unit is enabled as disclosed above regarding the bits of the cache control register, and its opcode or address at which it resides matches a value in the cache, and the control flag code as described in col. 3, line 15-16 has the "I" or "M" flag enabled as in col. 3, lines 30-35, the processor will in effect branch to that address in the cache memory, which contains the op-code to be inserted into the code stream) and to select said second memory address in response to said branch instruction if said mode indicator has a second value (when the cache unit is disabled, no additional code is inserted and the instruction is completed as normal, such as in col. 5, lines 66-67 and col. 6, lines 1-4; it is inherent

that a branch instruction includes some sort of address identifier to which the program flow will branch to).

37. Consider claim 24, Kasper discloses said set of code, when executed, tests for errors that occur during said run of said program (col. 1, lines 26-32, debug circuit; col. 2, lines 34-37, instruction content...may be corrected, modified and/or debugged; col. 5, lines 13-20, status buffer 34 of Figure 2 stores debug information), and wherein said set of code includes an instruction at one of said memory addresses; if a branch instruction is encountered and the cache unit is enabled as disclosed above regarding the bits of the cache control register, and its opcode or address at which it resides matches a value in the cache, and the control flag code as described in col. 3, line 15-16 has the "I" or "M" flag enabled as in col. 3, lines 30-35, the processor will in effect branch to that address in the cache memory, which contains the op-code to be inserted into the code stream).

38. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kasper as applied to claim 1 above, and further in view of Frank et al. (Frank) (US 5822578).

39. Consider claim 5, Kasper discloses said processing circuitry executes said set of code when said processing circuitry branches to said first address, and wherein said processing circuitry executes an instruction at said second address *before* completing execution of said set of code (col. 3, line 30, the I flag implies that the associated op-code in the cache is to be inserted into the instruction stream; col. 5, lines 62-65

disclose that the inserted code comes after the associated instruction). However, Kasper does not disclose it in the vice versa manner.

On the other hand, Frank discloses first executing a group of instructions, then executing the primary instruction (Figure 23, the instruction that causes trap is executed after the trap).

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that in certain situations, such as the trap conditions of Frank or the debugging conditions of Kasper, correct execution of a program can only be enabled when the additional, corrective instructions are executed before the catalyzing instruction and not after. Furthermore, it would have been readily recognized to one of ordinary skill in the art at the time of the invention that this change in when exactly the additional instructions are executed does not render the overall invention of Kasper unusable.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Frank with the invention of Kasper in order to correctly execute a program with corrective instructions in the case that doing so in the vice versa manner does not alleviate the errors of the specific program which is being debugged.

Conclusion

40. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Keith Vicary whose telephone number is (571) 270-1314. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

kv

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100